WHAT IS CLAIMED IS:

1. A method for dynamic load balancing resource allocation, comprising;

receiving a desired allocation of resources for servicing a plurality of consumer groups requests;

determining an actual allocation of the resources for a present operational period;

determining a temporary allocation of the resources for a next operational period relative to the desired allocation and the actual allocation;

allocating the resources to the consumer group requests in the next operational period according to the temporary allocation; and

selecting consumer group requests to be serviced by the resources based upon availability of the consumer groups requests and the amount of consumer groups requests being presently serviced.

2. A method as in claim 1, further comprising:

calculating a consumer load for each consumer group in response to the number of consumer groups requests being serviced, wherein each consumer group request is associated with a consumer group;

calculating a busyness factor for each resource in response to the number of requests being serviced; and

selecting the least busy resource to service the consumer group requests based on the consumer load and the busyness factor.

3. A method as in claim 1 wherein the actual resource allocation of consumer group requests is expressed in terms of a weight factor or in terms of a percentage.

4. A method as in claim 1 further comprising:

normalizing a sum of all resource allocations to one (1.0).

5. A method as in claim 1 wherein the actual resource allocation is defined in terms of a decay function of a factored sum of a measured current resource allocation percentage and an actual resource allocation value from a previous operational period.

6. A method as in claim 1 wherein the operational period is self-clocking or is a fixed time period.

7. A method as in claim 1 wherein the next operational period is adjusted inversely to a number of consumer group requests.

8. A method as in elaim 1 wherein the temporary allocation is expressed in terms of a requested resource allocation percentage, a rate of decay and an actual resource allocation percentage from a previous time period.

9. A method as in claim 1 wherein a priority for allocating the resources in the next operational period is determined using a weighted round robin scheme based on the temporary resource allocation percentage.

10. A method as in claim 9 wherein each consumer group request is associated with a consumer group, and wherein the weighted round robin scheme involves comparison of a weighted sum of serviced requests for each consumer group.

11. A method as in claim 10 wherein the consumer group with a lowest weighted sum is given the highest priority

12. A method as in claim 10 wherein the weighted sum comparison is made only among consumer groups with active requests outstanding.

13. A method as in claim 10 wherein a decay function is used with the weighted sum to minimize effects of an accumulated skewed request pattern.

14. A method as in claim 1 wherein the next operational period is shortened to minimize effects of an accumulated skewed request pattern.

15. A method as in claim 1 wherein restrictions are applied to servicing the consumer requests.

16. A method as in claim 2 wherein the calculation of the consumer group load for each group in a given operational period is based on a decay function of measured incoming request rate.

17. A method as in claim 16 wherein the measured incoming request rate is based on the ratio of the requests processed over the given operational period for a particular resource.

18. A method as in claim 2 wherein the busyness factor for a particular resource is based on a sum of all consumers loads on the particular resource.

19. A method as in claim 2 wherein each consumer group load is defined, for a given operational period, as a ratio of incoming consumer group requests divided by the serviced requests for a particular resource.

20. A computer readable medium embodying a computer program with code for dynamic load balancing resource allocation, comprising;

code for causing a computer to determine an actual allocation of the resources for a present operational period;

code for causing the computer to determine a temporary allocation of the resources for a next operational period relative to the desired allocation and the actual allocation;

code for causing the computer to allocate the resources to the consumer group requests in the next operational period according to the temporary allocation; and

code for causing the computer to select consumer group requests to be serviced by the resources based upon the amount of requests being presently serviced.

21. A computer readable medium as in claim 20 further comprising:

code means for causing the computer to calculate a consumer load for each consumer group in response to the number of consumer groups requests being serviced, wherein each consumer group request is associated with a consumer group;

code means for causing the computer to calculate a busyness factor for each resource in response to the number of requests being serviced; and

code means for causing the computer to select the least busy resource to service the consumer group requests based on the consumer load and the busyness factor.

22. A system for dynamic load balancing resource allocation, comprising:

a rcsource to be allocated for servicing consumer groups requests; and

a request arbitrator, including

means for determining an actual allocation of the resource for a present operational period,

means for determining a temporary allocation of the resource for a next operational period relative to the desired allocation and the actual allocation,

means for allocating the resources to the consumer group requests in the next operational period according to the temporary allocation, and

mcans for selecting consumer group rcquests to be serviced by thc resource based upon the amount of requests being presently serviced..

23. The system of claim 22 wherein there are at least two resources, and wherein the request arbitrator furthcr includcs

means for calculating a consumer load for each consumer group in response to the number of consumer groups requests being serviced, wherein each consumer group request is associated with a consumcr group,

means for calculating a busyness factor for each resource in rcsponse to the number of requests being serviced, and

means for selecting thc least busy resource to service the consumer group requcsts based on the consumer load and the busyness factor.

24. The system of claim 23 wherein the request arbitrator is further configured with means for kceping track of binding bctween consumer groups and resources.

25. The system of claim 23 wherein the request arbitrator is further configured with means for matching consumer group requests for a particular resource and its request queue.

26. The system of claim 23 wherein the request arbitrator is configured for being interrupt driven.

27. The system of claim 24 wherein the request arbitrator is further configured with

means for detecting a completion interrupt, and means, responsive to the completion interrupt, for identifying consumer groups requests having a particular binding and queuing them onto a request servicing queue.

28. The system of claim 24 wherein the request arbitrator is further configured with means for breaking an existing binding between a consumer group and the resource and for establishing a new binding.

29. A system for dynamic load balancing resource allocation, comprising:

a resource to be allocated for servicing consumer groups requests; and

a request arbitrator, including

logic operable to determine an actual allocation of the resource for a present operational period,

logic operable to determine a temporary allocation of the resource for a next operational period relative to the desired allocation and the actual allocation,

logic operable to allocate the resources to the consumer group requests in the next operational period according to the temporary allocation, and

logic operable to select consumer group requests to be serviced by the resource based upon the amount of requests being presently serviced..

30. The system of claim 29 wherein there are at least two resources, and wherein the request arbitrator further includes

logic operable to calculate a consumer load for each consumer group in response to the number of consumer groups requests being serviced, wherein each consumer group request is associated with a consumer group,

logic operable to calculate a busyness factor for each resource in response to the number of requests being serviced, and

logic operable to select the least busy resource to service the consumer group requests based on the consumer load and the busyness factor.